CLAIMS: *The following is a list of all claims in the application with their status and the text of all active claims.*

1.-15. (CANCELED)

16. (NEW) In a computer system, a method for viewing a large 3D model over a network (105), comprising:

a software renderer (103) running on a destination computer (101);

the software renderer (103) is in the active state of displaying a scene (700), and executing program_code (729), from a GX or GXML format file;

the software renderer (103) sends a request packet to a source computer (100) containing the logical name of a 3D model and subset selection criteria consisting of specific keyframe animations, polygon reduction, selecting polygons within a 3D volume space, or selecting textures and media;

the source computer (100) receives the packet and finds the 3D model data (mesh) and associated media files (images, videos, text) on a storage device (102);

the source computer (100) sends a packet containing a main header section (300) to the software renderer (103);

the source computer (100) sends a packet containing a mesh_resource_block_header (600) to the software renderer (103) for the selected 3D model data subset;

the source computer (100) sends packets containing mesh_resource_block_headers (600) to the software renderer (103) for logical parts of the unselected 3D model data subset, and the external_link (324) is set;

the source computer (100) sends a packet containing a block header section (301) for each of the associated textures and media to the software renderer (103);

the source computer (100) sends a packet containing the first part of a mesh_resource_data_block (1300) to the software renderer (103);

the software renderer (103) receives the packets in any order and renders the 3D model with associated textures and media; and

the software renderer (103) reads user input devices (such as mouse, keyboard, gamepad, joystick, etc.) that is handled by the program_code (729) to interact with the 3D model and view it from different viewpoints.

17. (NEW) Method according to claim 16, wherein:

GZIP compression is used on GX specific data (326 and 334), GXML, and text files;

JPEG or PNG compression is used on images;

UDP is used for packet transmission;

the request packet contains a bitrate_id, language_id, a screen_id, and a machine_id;

the software renderer (103) provides an API (Appendix B) to the program_code (729) the API containing at least the classes and functions to get and set the attributes of a data blocks section (302); and

the source computer (100) is running as a virtual computer, on a physical computer, in parallel with one or more other virtual source computers (100).

18. (NEW) Method according to claim 17, wherein the software renderer (103) runs as a web browser plug-in inside a web browser on the destination computer (101).

19. (NEW) Method according to claim 17, wherein the software renderer (103) runs as a computer game on the destination computer (101).

20. (NEW) Method according to claim 17, wherein the software renderer (103) runs inside a computer game on the destination computer (101).

21. (NEW) In a computer system, a method for playing a computer game over a network (105), comprising:

a software renderer (103) running on a destination computer (101);

the software renderer (103) sends a request packet to a source computer (100) containing the logical file name of a computer game;

the source computer (100) receives the packet and finds the game media files on a storage device (102);

the source computer (100) receives the packet and finds the game media files on a storage device (102);

the source computer (100) sends a packet containing a main header section (300) to the software renderer (103);

the source computer (100) sends a packet containing a scene_block_header (400) to the software renderer (103);

the source computer (100) sends a packet containing a scene_data_block (700) to the software renderer (103);

the software renderer (103) receives the packets in any order and executes the program_code (729) that is in Java byte code, assembler, or other code;

the software renderer (103) reads user input devices (such as mouse, keyboard, gamepad, joystick, etc.) that is handled by the program_code (729).

the program_code (729) loads 3D models and media from source computers (300) with effective content linking (figure 14); and

the program_code (729) renders the 3D models and media using hardware accelerated graphics libraries (such as DirectX and OpenGL).

22. (NEW) Method according to claim 21, wherein:

GZIP compression is used on GX specific data (326 and 334), GXML, and text files;

JPEG or PNG compression is used on images;

TCP is used for packet transmission of the scene;

UDP is used for packet transmission of program_code loading of 3D models and media from source computers (300) with effective content linking (figure 14);

the request packet contains a bitrate_id, language_id, a screen_id, and a machine_id;

the software renderer (103) provides an API (Appendix B) to the program_code (729) the API containing at least the classes and functions to get and set the attributes of a data blocks section (302); and

the source computer (100) is running as a virtual computer, on a physical computer, in parallel with one or more other virtual source computers (100).

23. (NEW) Method according to claim 22, wherein the software renderer (103) runs as a web browser plug-in inside a web browser on the destination computer (101).

24. (NEW) Method according to claim 22, wherein the software renderer (103) runs as a computer game on the destination computer (101).

25. (NEW) Method according to claim 22, wherein the software renderer (103) runs inside a computer game on the destination computer (101).

26. (NEW) In a computer system, a method for authoring computer games over a network (105), comprising:

a software renderer (103) running on a destination computer (101);

the software renderer (103) sends a packet containing a main header section (300) to the source computer (100);

the software renderer (103) sends a packet containing a mesh_resource_block_header (600) to the source computer (100) for a selected 3D model data subset;

the software renderer (103) sends packets containing mesh_resource_block_headers (600) to the source computer (100) for logical parts of a unselected 3D model data subset, and the external_link (324) is set;

the software renderer (103) sends a packet containing a block header section (301) for each of the associated textures and media to the source computer (100);

the software renderer (103) sends a packet containing the first part of a mesh_resource_data_block (1300) to the source computer (100); and

the source computer (100) receives the packets in any order and stores the 3D model on a storage device (102).

27. (NEW) Method according to claim 26, wherein:

GZIP compression is used on GX specific data (326 and 334), GXML, and text files;

JPEG or PNG compression is used on images;

UDP is used for packet transmission;

the request packet contains a bitrate_id, language_id, a screen_id, and a machine_id;

the software renderer (103) is in the active state of displaying a scene (700), and executing program_code (729), from a GX or GXML format file;

the software renderer (103) provides an API (Appendix B) to the program_code (729) the API containing at least the classes and functions to get and set the attributes of a data blocks section (302); and

the source computer (100) is running as a virtual computer, on a physical computer, in parallel with one or more other virtual source computers (100).

28. (NEW) Method according to claim 27, wherein the software renderer (103) runs as a web browser plug-in inside a web browser on the destination computer (101).

29. (NEW) Method according to claim 27, wherein the software renderer (103) runs as a software application on the destination computer (101).

30. (NEW) Method according to claim 27, wherein the software renderer (103) runs inside a computer game on the destination computer (101).